Implementation and Algorithms for Vertex-DSM-Tree

Falk Meissner and Eleanor Judd

March 31st 2006

I start with the Vertex DSM below, to document what's available in the Last DSM. For details how those bits get there refer to the other sections below.

1. Vertex DSM L1-VT201, layer2

All threshold bits of the Vertex tree are brought into the Vertex DSM and for ZDC they all get passed through to the last DSM. However, for BBC only the first threshold comparisons (th0) are given to the last DSM, since only one threshold bit is reserved in the TCU. The other threshold bits are here for easy later adjustments.
In parallel all three TAC differences are brought into the Vertex DSM. Windows are placed on the BBC small-tile and the ZDC TAC differences, and the "inside window" bits get passed through to the last DSM. The MSB of all 3 TAC differences are in the scaler output.
Note: The inclusion of the MTD causes some changes which are marked below

Inputs: From Small tile BBC-DSM BB101
(0-8) Small tile TAC-Difference
(12/13) Small tile ADC-East/West> th0
(14/15) Small tile ADC-East/West> th1

From Large tile BBC-DSM BB102
(0-8) Large tile TAC-Difference
(12/13) Large tile ADC-East/West> th0
(14/15) Large tile ADC-East/West> th1

From ZDC DSM ZD101
(0) ZDC1>ADC-th0E * dead time
(1) ZDC2>ADC-th0W * dead time
(2) ZDC1>ADC-th1E * dead time
(3) ZDC2>ADC-th1W * dead time
(4) time1-window-E * deadtime
(5) time2-window-W * dead time
MTD
(6) MTD bit
NO MTD
(6) att. Sum > th * dead time
(15-7) TAC-Difference

Registers: **L1-index: 0x19**
(Two independent timing windows for BBC small-tile and ZDC)
R0: SmallTile-ΔTMin0 (9)

R1: SmallTile-ΔTMax0 (9)
R2: SmallTile-ΔTMin1 (9)
R3: SmallTile-ΔTMax1 (9)
R4: ZDC-ΔTMin0 (9)
R5: ZDC-ΔTMax0 (9)
R6: ZDC-ΔTMin1 (9)
R7: ZDC-ΔTMax1 (9)
R8: BBC-SmallTile-Scaler-Select (3) **NOT Implemented**
R9: BBC-LargeTile-Scaler-Select (3) **NOT Implemented**
R10: ZDC-Scaler-Select (3) **NOT Implemented**

LUT: 1:1

Action

1st Clock:
Latch inputs
2nd Clock:
SmallTile-ΔTMin0 < Tacdiff < SmallTile-ΔTMax0 →BBC-tac0
SmallTile-ΔTMin1 < Tacdiff < SmallTile-ΔTMax1 →BBC-tac1
ZDC-ΔTMin0 < ZDC-Tacdiff < ZDC-ΔTMax0 →ZDC-tac0
ZDC-ΔTMin1 < ZDC-Tacdiff < ZDC-ΔTMax1 →ZDC-tac1
Define photon-pretrigger = Small-tile East ADC > th0 AND Small-tile West ADC > th0
3rd Clock:
Empty/delays/mapping
4th Clock:
Latch Outputs

Output: **THIS IS WHATS AVAILABLE IN THE LAST DSM!**

(0) BBC-tac0	TAC windows
(1) ZDC-tac0	
(2) SmallTile-ADC-th0-East	Small Tile BBC ADC th0
(3) SmallTile-ADC-th0-West	
(4) LargeTile-ADC-th0-East	Large Tile BBC ADC th0
(5) LargeTile-ADC-th0-West	
(6) ZDC-ADC-th0-East	ZDC ADC th0
(7) ZDC-ADC-th0-West	
(8) ZDC-TAC-win-East	ZDC TAC window
(9) ZDC-TAC-win-West	
MTD	
(10) MTD bit	MTD trigger
NO MTD	
(10) ZDC-attSum-th0	ZDC attenuated Sum
(11-13) spare	
(14) ZDC-ADC-th1-East	ZDC ADC th1
(15) ZDC-ADC-th1-West	

Scalers: (0) BBC-tac0	TAC windows
(1) BBC-tac1	

(2) ZDC-tac0
(3) ZDC-tac1
2005 Algorithm (L1_VT201_TAC)
(4-7) 4 MSB of Small Tile TAC-Difference TAC differences
(8-10) 3 MSB of Large Tile TAC-Difference
(11-14) 4 MSB of ZDC TAC-Difference
Temporary 2006 Algorithm (L1_VT201_2006a)
(4-7) 4 MSB of ZDC TAC-Difference
(8-10) 3 MSB of Large Tile TAC-Difference
(11-14) 4 MSB of Small Tile TAC-Difference
(15) Photon-pretrigger

2. BBC BB001-006; layer0;

Input: 8*8bits inner PMTs ADC
8*8bits inner PMTs TAC
(Sorted by quadrants, T/B(3PMTs), S/N(5PMTs)

LUT: timing adjustments/pedestal subtraction for each PMT

Registers: **BBC; index 0x10, 0x12, 0x14, 0x16, 0x18, 0x19**
R0: PMT-ADC-Threshold (8)
R1: TACWindowMin (8)
R2: TACWindowMax (8)
R3: Deadtime (4)

1st Latch Inputs

2nd for each PMT:
Timing Window: TACWindowMin<TAC<TACWindowMax →Good-TAC
ADC above threshold: ADC> PMT-ADC-Thresh →Good-ADC
Delay ADC values by 2 clocks until 4th
Delay TACs by 3 clocks until 1st override

3rd Delay Good-TAC bits by one clock until 4th
Apply deadtime to Good-ADC bits →Good-ADC+KillerBit
Instantiate deadtime components

4th Define 'good hit'= ADC above th and dead time ok, TAC window ok
Gate ADC values: 'good-hit' gated ADC =input ADC
All other ADCs set to '0' to exclude them from ADC sum
Build quadrant hit map bits from good hits
Quad-Hit-Map (4 bits): Two Quads per DSM board*(inner/outer circle of small tile Annulus).
Watch: North and South PMT are swapped for East and West
PMT 1—7—8 →Top
PMT 4—12—13 →Bottom
PMT 5—6—14—15—16 →East/South or West/North
PMT 2—3—9—10—11 →West/South or East/North

1st override: Delay quad-hit-map by 3 cycles until 4th override (output mapping)

Sum gated ADC values intermediate step (sum ADC0-3 and ADC4-7)
Compare 4*2TACs
Only two TAC values at a time can be compared per clock cycle. For 8 TACs, 3 cycles are needed →have to use override cycles.
If there are no good PMT hits at all, TAC quality bit is set to zero.

2nd override: Total ADC Sum (0-7)
Compare 2*2TACs

3rd override: Delay ADC sum by one cycle
Compare 2 TACs →fastest TAC

4th override: Map output

Output: (0-10) ADC-Sum
(11-14) Quadrant-hit-map (2 Quads*Inner/Outer)
(15) Spare
(16-23) Max TAC
(24) TAC Quality bit
(25-31) Spare

3. BBC BB101-2, layer1

Inputs: from 4 small tile ADC/TAC boards (2 East/2West) and 2 large tile ADC/TAC boards (1 East/West)
(0-10) ADC-Sum
(11-14) quad-hit-map
(16-23) Max TAC
(24) TAC quality

Registers: **BBC; index: 0x1a, 0x1b**
Four registers, all thresholds can be set independently
R0: ADC-Thresh0-East (11)
R1: ADC-Thresh0-West (11)
R2: ADC-Thresh1-East (11)
R3: ADC-Thresh1-West (11)

LUT: 1:1

1st Clock: For East and West separately:
Compare 2 TACs →Fastest TAC
Sum ADC values

2nd Clock Calculate TAC-East-TAC-West →TAC-Difference
Sum-ADC-East > ADC-Thresh0-East
Sum-ADC-West > ADC-Thresh0-West
Sum-ADC-East > ADC-Thresh1-East
Sum-ADC-West > ADC-Thresh1-West
Combine Quad-Hit-Maps: East/West/Inner/Outer

Output: (JP6-upper bits of output) to VT201
(0-8) TAC-Difference
(9-11) empty
(12-13) ADC-East/West > th0
(14-15) ADC-East/West > th1

Scalers: (JP1 lower bits of output)
(0-3) quad-hits-East T/B/N/S
(4-7) quad-hits-West T/B/N/S
(8/9) hits-in-inner circle East/West
(10/11) hits-in-outer circle East/West
(12-13) ADC-East/West > th0
(14-15) ADC-East/West > th1

4. ZDC ZD001, layer0

Input: (only the used one are listed)

zdc-adc-East (8)	input 1 (15-8)
zdc-adc-West (8)	input 3 (15-8)
zdc-tac-East (8)	input 7 (15-8)
zdc-tac-West (8)	input 7 (7-0)
zdc-attSum-East+West (8)	input 4 (15-8)

Registers: **BBC; index: 0x1e**
(ADC thresholds th0/th1 independent for East/West)

R0: zdc-adc-East-th0 (8)	
R1: zdc-adc-West-th0 (8)	
R2: zdc-adc-East-th1 (8)	
R3: zdc-adc-West-th1 (8)	
R4: zdc-tac-East-Min (8)	
R5: zdc-tac-West-Min (8)	
R6: zdc-th0-timegap (4)	deadtime for ADC E/W>th0
R7: zdc-th1-timegap (4)	deadtime for ADC E/W>th1
R8: zdc-timewin-timegap (4)	deadtime for TAC-window
R9: zdc-attSum-th (8)	
R10:zdc-attSum-timegap (4)	deadtime for attSum>th
R11:zdc-tac-East-Max (8)	
R12:zdc-tac-West-Max (8)	

LUT: Pedestal subtraction

1st. Clock: compare thresholds: zdc-adc*>th*
tac-timewindows: min <= zdc-TAC-East/West < max
Calculate TAC difference, no quality cuts here.

2nd Clock: force deadtime for the timegap*RHIC crossings

Output: (0) zdc1 > ADC-th0E * dead time R6*104ns
(1) zdc2 > ADC-th0W * dead time R6*104ns
(2) zdc1 > ADC-th1E * dead time R7*104ns

(3) zdc2 > ADC-th1W * dead time R7*104ns
(4) time1-window-E * deadtime R8*104ns
(5) time2-Window-W * dead time R8*104ns
(6) att. Sum > th * dead time R9*104ns
(15-7) TAC-Difference

5. MTD MD001, layer0

Input: 10 ADC values from the muon detectors and the over-lapping CTB trays.
Ch 0:5 – CTB Trays 47-1, 48-1, 49-1, 67-1, 68-1 and 69-1
Ch 6:7 – MTD1-1 and MTD1-2
Ch 8:9 – MTD2-1 and MTD2-2

Registers: **BBC; index: 0x1c**
R0: ADC threshold for defining a hit

LUT: Pedestal subtraction

Action

1st Clock:
Latch inputs
2nd Clock:
Compare all 10 inputs to the threshold in R0
3rd Clock:
Combine the threshold results:
(ch0 or ch1 or ch2 or ch3 or ch4 or ch5) AND
(ch6 or ch7) AND (ch8 or ch9)
4th Clock:
Latch output

Output: (0) MTD bit
(15-1) unused

6. ZDC ZD101, layer1 – MTD Version

Inputs: from ZD001
(0) zdc1>ADC-th0E * dead time R6*104ns
(1) zdc2>ADC-th0W * dead time R6*104ns
(2) zdc1>ADC-th1E * dead time R7*104ns
(3) zdc2>ADC-th1W * dead time R7*104ns
(4) time1-window-E * deadtime R8*104ns
(5) time2-Window-W * dead time R8*104ns
(6) att. Sum > th * dead time R9*104ns
(15-7) TAC-Difference

From MD001
(0) MTD bit
(15-1) unused

Registers: None

LUT: 1:1

Action

1st Clock:
Latch Inputs
2nd Clock:
Map input to output and scaler
Substitute the MTD bit for the ZDC E+W attenuated sum bit (bit 6)
3rd Clock:
Delay
4th Clock:
Latch Outputs

Output: (JP6 bits 31-16 of output) to VT201
(0) zdc1 > ADC-th0E * dead time R6*104ns
(1) zdc2 > ADC-th0W * dead time R6*104ns
(2) zdc1 > ADC-th1E * dead time R7*104ns
(3) zdc2 > ADC-th1W * dead time R7*104ns
(4) time1-window-E * deadtime R8*104ns
(5) time2-Window-W * dead time R8*104ns
(6) MTD bit
(15-7) TAC-Difference

Scalers: (JP1 bits 15-0 of output) to Scaler Source patch Panel
(0) zdc1 > ADC-th0E * dead time R6*104ns
(1) zdc2 > ADC-th0W * dead time R6*104ns
(2) zdc1 > ADC-th1E * dead time R7*104ns
(3) zdc2 > ADC-th1W * dead time R7*104ns
(4) time1-window-E * deadtime R8*104ns
(5) time2-Window-W * dead time R8*104ns
(6) MTD bit

7. Tile-PMT-DSM Input Mapping

BBC scintillator tiles numbers are specified at http://www.star.bnl.gov/STAR/html/bbc_l/geom/front_view.html which shows the BBC scintillator array from a vantage point which is outside of the STAR magnet toward the IP. The numbering scheme applies to the East and West sides of STAR. (Note: this viewpoint is contrary to the usual star definitions looking from the IP toward East and West.) The Tile-to-PMT mapping comes from Les' email. The PMT-to-DSM channel assignments are from the trigger page cable map http://www.star.bnl.gov/STAR/html/trg_l/TSL/Schematics/BBC_Crate_Cable_Map.htm

7.1. Small Tiles

Tile	PMT	East	Pos.	West	Ring	DSM E/W	ADC In / TAC In
1	1		T		inner	BB001/002	JP2-ch0l / JP4-ch4l
2	2	N		S	inner	BB001/002	JP7-ch1h / JP9-ch5h
3	3	N		S	inner	BB001/002	JP3-ch2l / JP5-ch6l
4	4		B		inner	BB003/004	JP2-ch0l / JP4-ch4l
5	5	S		N	inner	BB003/004	JP7-ch1h / JP9-ch5h
6	6	S		N	inner	BB003/004	JP3-ch2l / JP5-ch6l
7	7		T		outer	BB001/002	JP2-ch0h / JP4-ch4h
9			T		outer		
8	8		T		outer	BB001/002	JP7-ch1l / JP9-ch5l
10	9	N		S	outer	BB001/002	JP3-ch2h / JP5-ch6h
11	10	N		S	outer	BB001/002	JP8-ch3l / JP10-ch7l
12	11	N		S	outer	BB001/002	JP8-ch3h / JP10-ch7h
13	12		B		outer	BB003/004	JP2-ch0h / JP4-ch4h
15			B		outer		
14	13		B		outer	BB003/004	JP7-ch1l / JP9-ch5l
16	14	S		N	outer	BB003/004	JP5-ch2h / JP5-ch6h
17	15	S		N	outer	BB003/004	JP8-ch3l / JP10-ch7l
18	16	S		N	outer	BB003/004	JP8-ch3h / JP10-ch7h

7.2. Large Tiles

Tile	PMT	East	Pos.	West	Ring	DSM E/W	ADC In / TAC In
19	17		T		inner	BB005/006	JP2-ch0l / JP4-ch4l
20	18	N		S	inner	BB005/006	JP2-ch0h / JP4-ch4h
21		N		S	inner		
22	19		B		inner	BB005/006	JP7-ch1l / JP9-ch5l
23	20	S		N	inner	BB005/006	JP7-ch1h / JP9-ch5l
24		S		N	inner		
25	21		T		outer	BB005/006	JP3-ch2l / JP5-ch6l
26			T		outer		
27			T		outer		
28	22	N		S	outer	BB005/006	JP3-ch2h / JP5-ch6l
29		N		S	outer		
30		N		S	outer		

31	23		B	outer	BB005/006	JP8-ch3l / JP10-ch7l
32			B	outer		
33			B	outer		
34	24	S	N	outer	BB005/006	JP8-ch3h / JP10-ch7l
35		S	N	outer		
36		S	N	outer		

8. BBC-DSM tree

The basic idea is to compare the TAC and the ADC value individually for each PMT. There are four layer0 DSM boards for the small tiles, and two layer0 DSM boards for the large tiles, each with 8 ADC and the 8 corresponding TAC channels. Two cables connect each small tile layer0 board to their layer1 DSM board (making the max of 8). The large tile layer0 DSM boards have a separate DSM board in layer1. Large and small tiles are combined in the Vertex DSM.

TAC and ADC are compared for each PMT separately. Only ADC values with a good TAC make it into the ADC sum. Only TACs with ADC values above threshold make it into the fastest TAC race.

One DSM has 8 input channels, that means each small tile DSM has all PMTs of two quadrants (Top(3) + North(5)) or (Bottom(3)+South(5)). We have separate bits for each of the 16 sub quadrants: (East/West)*(Large/Small)*(Top/Bottom/North/South). The small tile quadrant hits go into the scaler. The FPGA code is the same for all 6 layer0 DSM boards (BB001-6), even though the quadrant information does not make sense for the large tiles.

Large and small tiles are treated as separate detectors. There are only 16 bits available from BB101 and BB102 to the Vertex DSM, the possibilities to combine large and small tile information separately for East and West are limited. Specifically we do not have:
a total ADC sum. There are only separate bits for large> threshold and small > threshold.
a combined Quadrant-hit-pattern separately for East and West which combines small and large tiles, i.e. T/B/S/N -(Large OR Small).

Thanks to Jack, the (modified) DSM tree is cabled and documented

9. Definitions

Quadrants: consist of either 3PMTs(Top/Bottom) and 5(North/South). The Tile/PMT numbers are swapped for East and West so that East:North=West:South and vice versa.
Small Tiles: #1-18; Large Tiles: #19-36 Inner/Outer Ring within the small or large annulus: Small Tiles: Inner #1-6; Outer #7-18 Large Tiles: Inner #19-24; Outer #25-36
Clock Cycles: There are four DSM board clock ticks per RHIC cycle. First and Fourth are needed to latch in/out the data. The second and third are available for calculations.

We can have one major serial operation per clock cycle on a channel. But one can have them in parallel on different channels. These major operations are for instance: Comparison to a threshold, Summing 8*8 bit numbers, (summing16*8 bits has to be done in stages takingtwo cycles), Combining several and/or operations.
Override Cycles: To extend the limit of 2clock cycles we can use override cycles. Here we use two RHIC cycles to process input. The output of cycle ’n’ corresponds to input from ’n-1’.This needs to be done in all DSMs in the affected layer ofthe tree (e.g BB001/2/3/4) to keep the output of that layer consistent. We have enough empty synchronization/delaycycles in the trees to implement this.An example for using overrides is the killer bit algorithm, where the condition if a hit in cycle ’n’ may be good is calculated in cycle ’n-1’. Nevertheless, this is a more involved and messy feature.
13
Max TAC The TAC is the time difference of the leading edge of the signal to a common STOP signal. Therefore, the fastest(earliest signal) is the largest number =Max TAC. Each Tac value has a quality bit, i.e. the corresponding ADC value is above a certain thrshold. Only TACs with the quality bit set are compared.